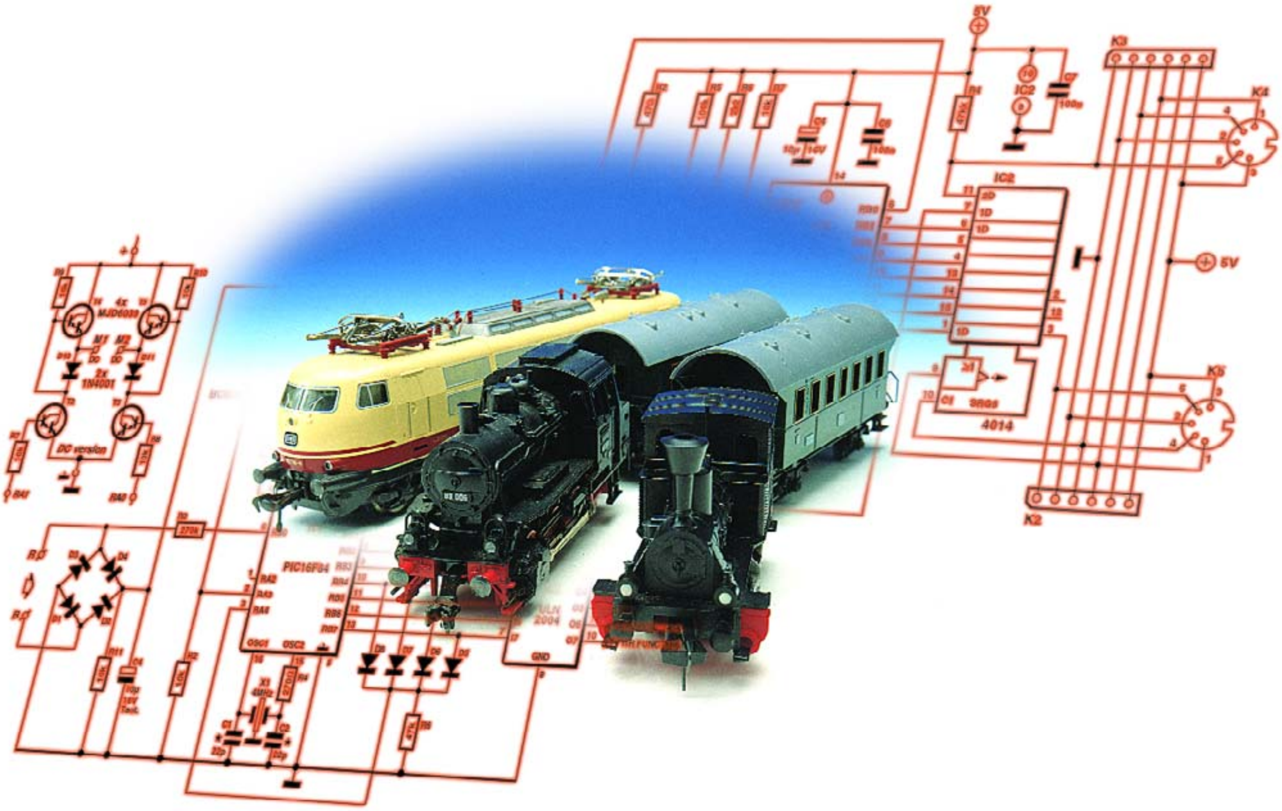
# EEDTS Pro
# Super Loco Decoder Upgrade

## with a return signaller upgrade

Design by H. Prince

When the announcement was made for a new version of the PC software and controller for the popular EEDTS Pro model train control system it was inevitable that an upgrade for the super loco decoder would soon follow. The original super loco decoder was published in the October 1999 issue of Elektor Electronics. The version described here replaces the old one and offers several enhancements, such as a programmable speed table.



The new version of the super loco decoder (ESLD) doesn't just have extra functions to support the new PC software. Improvements have also been made to some of the original functions.

It now takes into account which type of motor is used in the loco. There are trains that use a.c. or d.c. motors. The properties of d.c. motors are different to those of a.c. motors, which causes d.c. locos to run too fast at lower speed settings. Because the use of d.c. motors is increasing, the speed table can now be adapted to suit the properties of the motor.

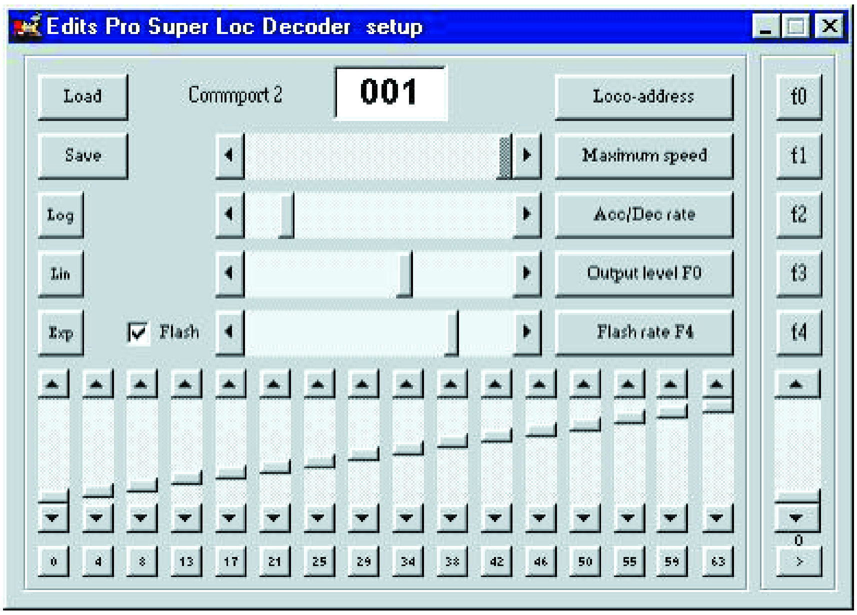


Figure 1. Screen dump of the program for setting up the new super loco decoder.

## The speed table

The speed of the motor can be adjusted in 16 steps. Each position of the speed controller corresponds to a location in a table that is stored in the EEPROM of the decoder. This table contains values that determine how fast the loco runs. In principle, location 0 (de lowest position of the controller) contains the value 0 and location 15 contains the largest value of 0FFh.

The decoder was originally designed for use in Märklin locos with a.c. motors and the table has been defined such that a relatively high voltage is fed to the motor at lower settings of the controller.

Locos using a d.c. motor normally have a permanent magnet and higher gearing. This causes them to start sooner and makes them run too fast at low speed settings. This can be partially offset by reducing the maximum speed, but that is not an ideal solution. For this reason we have changed the firmware such that the contents of this table can be modified.

One of the original design criteria for programming the settings into the decoder was that it should be possible to do so using only simple tools. A constraint of this is that only 'trits' from the Motorola format (which the protocol is based on) can be used. Since two trits are used for the commands, the maximum number of possible commands is $3^2 = 9$. Of these, seven are used for common settings (**Table 1**). The other two are therefore available for the programming of the speed table. Since use is made of the standard Motorola format this can also be done with the hand-programmer described in the original article.

We realise that programming a complete table would be a time consuming job. That is why we've introduced a utility program that can be found on the EEDTS Pro website (www.gironet.nl/home/editspro) and on the Elektor website (www.elektor-electronics.co.uk, order code **020094-11**).

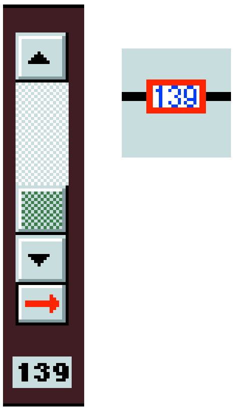


Figure 2. Speed control for a loco and the return button with an address of 139.

## Programming

In order to give enthusiasts the chance to program the decoder with their own programmers, we'll explain how the table (as well as the other parameters) can be programmed. It should be well known that the information put on the rails consists of four trits for the loco address, one trit for the front lights and four trits for the speed data. Further details regarding the Motorola protocol can be found in various articles in *Elektor Electronics* in 1999.

For setting up an ESLD a fixed address (79) has been reserved, which obviously requires four trits. The other five trits are used to convey setup information, such as loco address, maximum speed, rates of acceleration and deceleration, etc. Each command has to be correctly received by the decoder a minimum of ten times before it is accepted. In practice you will find that when a command has been put on the rails for two seconds it will certainly have been accepted.

For completeness we have shown all possible settings in **Table 1**.

The procedure for programming the table is simple: first the address of the location that we wish to modify is programmed, and then the actual value is programmed in that location. This way the locations can be programmed one by one.

## The software

To make life easier, a software application was written to help set up an ESLD. This can only be used in combination with the EEDTS Pro controller. Users of other types of controller will have to write their own program.

When the program is started, a window appears that contains all settings. Several of the buttons are familiar from the setup screen in EEDTS Pro: loco-address, maximum speed, acc/dec rate, brightness and flash rate. To the left of these are the accompanying slider controls.

Underneath these are the controls for setting up the table values. Each slider control corresponds to a location in the table and a setting of the speed controller. The setting for each slider control is shown in a button underneath it. When this button is clicked (with the left mouse button) the value of the corresponding control will be programmed into the memory of the decoder. This has to be done once for each of the locations.

## Table 1. Programming the settings.

| 987654321 | Trit |
|---|---|
| LLLL0 | program loco address (LLLL = 1 - 255) |
| MMM01 | program maximum speed (MMM = 0 - 15) |
| AAA0X | program rate of acceleration & deceleration (AAA = 0 - 15) |
| LLLL0 | program loco address |
| SSS11 | program brightness of lights for function 0 (SSS = 0 - 15) |
| KKK1X | program flashing rate for function 4 (KKK = 0 - 26) |
| LLLL0 | program loco address |
| CCCX1 | **new: program table address (CCC = 0 - 15)** |
| EEEXX | **new: program speed (EEE = 0 - 255)** |

Where: trit 0 = 00 binary, trit X = 01 binary, trit 1=11 binary; trit 1 - 4 = loco address 79

Since the amount of information per loco has become rather large, and it's not possible to read information back from the decoder, the 'Save' and 'Load' load buttons have been added to store the data to the hard disc and subsequently retrieve it. The settings are stored in a file with a name the same as the number in the top input box and an extension of '.stp'. Underneath the 'Save' button are three buttons that have been pre-set with standard table values: logarithmic, linear and exponential. These can be used as a starting point for creating your own tables.

The program uses the same setup file (com.inf) for the COM port as EEDTS Pro and will therefore start up using the same port that EEDTS Pro is set up for, as long as the programs are in the same directory.

## Tabel 2.

### Addresses in the return signaller.

| Loco addr. | Return button | Bit 7 |
|---|---|---|
| Without IR | blank (00) | |
| 0 (*1 | not permitted | |
| 1 | 1 | 1 |
| ⋮ | ⋮ | ⋮ |
| 78 | 78 | 1 |
| 79 (*1 | not permitted | - |
| 80 | 80 | 1 |
| 81 | 81 | 1 |
| 82 | 82 | 1 |
| 83 | 83 | 1 |
| ⋮ | ⋮ | ⋮ |
| 127 | 127 | 1 |
| 128 (*2 | not permitted | 0 |
| 129 | 129 | 0 |
| ⋮ | ⋮ | ⋮ |
| 254 | 254 | 0 |
| 255 | 255 | 0 |

(*1 This address is blocked in the ESLD
(*2 The loco will accept this address, but the return button will be inactive

Another COM port can be selected by double clicking on the text. The software is available only in English.

*Remember that when programming the decoder, all locos on the rails at that time will take the new settings. The programming is therefore not determined by the loco address on the screen.*

On the far right are several buttons that are used to test the settings just made. The loco with the same address as that in the input box for the loco-address will respond to these. In this way the speed, the direction and the five functions can be operated.

## Other enhancements.

The new version of the EEDTS Pro control software greatly extends the loco address range to 255. A consequence of this is that both the loco decoder as well as the address return signaller have to be made compatible with this. Although the ESLD was originally designed for an address range from 1 to 81, the old version could handle some addresses above 81. Since we were already making changes to it, we decided to make the new decoder recognise the full range of addresses.

Several users have managed to program a loco with an address of 79. The result of this is that the loco will run, but can never be set to another address. The new firmware blocks this address.

## Return signaller

Obviously the loco address return signaller (Elektor Electronics January 2000) also has to recognise the higher addresses. In the original design the decoded data was placed on the bus as soon as the IR information was received. Only when the contact rail was closed would bit 7 be set high and the data would be read by EEDTS Pro. That means that 129 (81h) corresponds to loco address 1, 130 (82h) to address 2 etc. In the new version this is dealt with differently. The return signaller will only place the data onto the bus when the contact rail is closed. To keep the return signaller compatible with the previous version, the lower range remains unchanged, but is extended to 127 (0FFh) and the rest of the range (loco addresses 128-255, 00h-07Fh) has been added to that. A consequence of this is that loco-address 128 (80h) is seen by EEDTS Pro as address 00h, which is then recognised as a loco without infrared return signalling.

This is shown more clearly in **Table 2**, along with other restrictions.

We would like to point out that both versions of the ESLD and return signaller can be used simultaneously, even with the old version of EEDTS Pro. It is only with the new version of EEDTS Pro that the extended addressing range becomes available. Since only the firmware has changed, the PCBs for the decoder and the return signaller have remained the same.

(020094-1)

**Order codes for PICs:**

Programmed PIC for the super loco decoder: 020094-41

Bulk packs are also available for 5, 10 and 20 pieces including PCB (refer to the Readers' Services pages)

Programmed PIC for the return signaller: 020095-41